Universität Karlsruhe - Fakultät für Informatik - Bibliothek - Postfach 6980 - 76128 Karlsruhe

Auction-Based Traffic Management : Towards Effective Concurrent Usage of Road Intersections

Heiko Schepperle, Klemens Böhm

Interner Bericht 2008-1
(Technical Report 2008-1)

Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

ISSN 1432-7864

Fakultät für **Informatik**

Auction-Based Traffic Management: Towards Effective Concurrent Usage of Road Intersections*

Heiko Schepperle, Klemens Böhm
Universität Karlsruhe (TH), Germany
{schepperle,boehm}@ipd.uni-karlsruhe.de

Abstract

Valuation-aware traffic-control mechanisms for road intersections take the valuations of reduced waiting time of the individual drivers into account. They use agents to avoid any disturbance of the driver, and they feature mechanisms specifically designed for negotiating valuations. While such mechanisms do indeed increase overall driver satisfaction, they only allow one vehicle at a time to use the intersection so far. But the concurrent usage of an intersection is an important feature needed in reality. This paper proposes new auction-based intersection-control mechanisms allowing for concurrency. Their design has been challenging due to complex traffic-specific requirements like the high number of agents involved, or varying space and time consumption of vehicles crossing the intersection. Experiments show that they yield a significantly higher degree of overall driver satisfaction than state-of-the-art mechanisms.

1 Introduction

Valuation-aware traffic-control mechanisms for road intersections take the valuations of reduced waiting time of the individual drivers into account [12, 13]. For instance, think of a salesman driving to a customer. He would be willing to pay some money to arrive earlier. Another driver would only be willing to pay less because his valuation is lower. Respective systems typically comprise driver-assistance systems which exchange the valuations of their drivers with other vehicles or the infrastructure.

Our goal is the design of *effective* mechanisms for intersection control, i.e., mechanisms which reduce the average waiting time weighted by the driver valuations. Existing solutions either support concurrent usage of the intersection but are not valuation-aware, e.g., traffic lights or reservation-based systems [1, 2], or are valuation-aware but do not feature concurrency, e.g., [12, 13]. Assuming no concurrency at intersections is not realistic. Thus, the combination of valuation-awareness and concurrency in intersection control is an open issue and promises a significant reduction of average weighted waiting time.

Any intersection-control system mediates between road users with opposing goals. The system offers *time slots*, i.e., the privilege to cross the intersection during a certain period of time, to road users. The valuation-aware system comprises a mechanism specifying how road users can negotiate a time slot. The design of such mechanisms is important because they affect negotiation strategies and the outcome of the system as a whole [11]. Because valuations are private information, a market approach is appropriate [8]. Auctions are suitable to identify the drivers with the highest valuations [8], and we focus on auction-based mechanisms using software agents. We propose two new auction-based mechanisms for intersection control allowing for concurrency, *Free Choice* and *Clocked*. *Free Choice* lets the winning vehicle freely choose a time slot within an interval, while *Clocked* auctions intervals whose duration is the one of a time slot. To our knowledge, they are the first 'location-aware' auction mechanisms, i.e., auctions where the position of the participants plays a role.

The design of valuation-aware intersection-control mechanisms is challenging. One reason is that vehicles are physical objects and subject to restricting physical laws, e.g., *speed and acceleration limitations*. Concur-

---

[*] This work is part of the project DAMAST (Driver Assistance using Multi-Agent Systems in Traffic) (http://www.ipd.uni-karlsruhe.de/%7Edamast/) which is partially funded by init innovation in traffic systems AG (http://www.initag.com/).

rency adds to the complexity: In particular, the mechanisms should yield *a high degree of concurrent usage* of the intersection and at the same time *prevent vehicles from colliding*. Given that vehicles with different valuations and different speeds may arrive at the intersection at any time, this requirement is far from trivial. Further, auctions should be designed such that they do not degenerate, e.g., they should have more than one participant. One-participant auctions tend to occur if the auctions are executed immediately one after another.

We compare both mechanisms to a state-of-the-art mechanism for intersection control which uses agents as well but is not valuation-aware. We determine their effectiveness for different vehicle densities and different degrees of concurrency. Our evaluation shows that both the number of vehicles and the degree of concurrency influence the effectiveness. *Free Choice* always reduces average weighted waiting time (up to 38.1%). With *Clocked*, this is only true for lower degrees of concurrency and high demands.

This project is part of a larger effort where, together with a supplier of intelligent transportation systems, we explore the opportunities and bounds of market-based traffic management. Naturally, this paper investigates one specific problem and leaves aside other ones, e.g., legacy issues, i.e., old vehicles or pedestrians without the equipment assumed here. We also assume that vehicles cannot overtake each other in the vicinity of an intersection; this is in line with traffic regulations in many countries. Another assumption is that robust and cheap mobile communication mechanisms will be available in the future. Further, this paper does not consider cooperation among different intersections. These assumptions are realistic, at least when looking at the future. Today, they already hold for some closed traffic areas such as transship areas of seaports or company premises.

Paper outline: Section 2 discusses related work and Section 3 preliminaries. We describe our new mechanisms in Section 4. Section 5 is our evaluation and Section 6 concludes.

2 Related Work

We discuss agent-based approaches for traffic management which we deem important for our work.

The reservation-based mechanism of [1, 2] is the first agent-based intersection-control mechanism for road traffic relying on data provided by vehicles. It uses a first-in first-out strategy and does not feature valuation-awareness. An extension [3] introduces priorities for emergency vehicles. Traffic-signal priority system, e.g., [6], prioritize public transportation. [3, 6] prioritize only a small fraction of the overall traffic and ignore the valuations of other road users. The first valuation-aware mechanisms for road-traffic control are proposed in [13]. There, driver-assistance agents can exchange the time slots that have been allocated to them. [12] compares an auction mechanism for time-slot allocation to a first-in first-out mechanism. In both approaches the scenario is limited: There is no support for concurrency. [12, 13] also do not investigate how valuation-aware mechanisms perform with different vehicle densities.

Next to agent-based mechanisms for traffic control at road intersections, there are related agent-based approaches for air-traffic control. [7] uses agents for arrival, departure and gate planning in air traffic. If a plan has to be changed, agents representing the airlines negotiate plan-repair schemes. The time frame for planning is several days. In road traffic, it is seconds to a few minutes at best. Therefore, an initial plan has to be negotiated automatically while driving, and deviations from the plan are more unlikely. Thus, road traffic calls for an effective initial allocation of time slots, unlike plan repair.

[9] uses agents to plan the deicing of airplanes at airports. Agents representing the planes negotiate time slots at the deicing stations. The authors propose two mechanisms: a valuation-aware Vickrey auction and a reservation-based mechanism with decommitment penalties. The authors conclude that auctions are more efficient but less fair, i.e., waiting times have a larger standard deviation. The approach is similar to [12] and has the same drawbacks: Only one plane is deiced per time, and deicing takes the same time for every plane. The scenario is less complex than in [12] because the order of arrival of planes does not matter.

Summing up, realistic valuation-aware mechanisms for traffic control at road intersections are missing. Thus, we need such mechanisms which allow vehicles to cross an intersection in parallel without interfering with each other. These mechanisms must be effective, i.e., they must reduce the average weighted waiting time.

3 Preliminaries

Underlying Definitions. Incoming and outgoing lanes are part of the *neighborhood* of an intersection. *Inter-*

section lanes are the lanes which cross the intersection and turn into various directions. Vehicles approach the intersection on *incoming lanes* and leave it on *outgoing lanes*. Intersection lanes are *conflicting* if they touch or intersect each other. We refer to the area used by two different intersection lanes as *conflict area*.

The *overpass time of a trip* is the time a vehicle would need for the entire trip if it was the only vehicle in the neighborhood, i.e., no obstruction due to other vehicles or traffic-control mechanisms. By definition, the *waiting time* of a vehicle is the difference of the actual travel time and the overpass time. A driver has a *valuation* for reducing his waiting time. In the following, this valuation is the amount of money one would be willing to pay if his waiting time was reduced by one second. While other valuation functions are conceivable as well, we limit ourselves to linear ones in this paper. The *weighted waiting time* is the waiting time of a driver multiplied with his valuation. In our experiments, we will give much importance to the *average waiting time* (*avgWT*) and the *average weighted waiting time* (*avgWWT*) of all vehicles. The *minimum approaching time* $t_{approach,min}$ is the minimum time a vehicle needs to drive from the beginning of the neighborhood where it can request a time slot to the beginning of the intersection where it can use the slot. Our objective is not to increase the benefit of the operator of an intersection, so we focus on *effectiveness* i.e., reducing avgWWT. This corresponds to increasing the overall satisfaction.

The *demand* is the average number of vehicles arriving on an incoming lane per hour. With n incoming lanes, the intersection has to deal with n times the demand. With intersection-control mechanisms, a driver can earn or spend money. His *utility* is the monetary balance minus the weighted waiting time.

Example 1 *Let the valuation of a driver per second be* 0.1. *If he has to wait* 20 *seconds, his utility is* $-20 * 0.1 = -2$. *If he procures another time slot by spending* 0.5, *and if this exchange brings down his waiting time to* 12 *seconds, his utility increases to* $-0.5 - 12 * 0.1 = -1.7$.

We distinguish three degrees of concurrency. We call a setting where only one vehicle can cross the intersection per time *intersection exclusive* (*IE*), i.e., concurrency is not supported. We refer to a situation where vehicles can cross the intersection concurrently only if the intersection lanes used are not conflicting *lane exclusive* (*LE*). A setting where vehicles can cross the intersection concurrently as long as no conflict area contains more than one vehicle is *conflict area exclusive* (*CAE*). *CAE* is the highest possible degree of concurrency and is less restrictive than *LE*. With *LE* in turn, vehicles have to wait until the preceding vehicle on the same incoming lane has left the intersection. Time slots are *conflicting* if they cannot be used simultaneously, according to the degree of concurrency used.

Architecture. To avoid distraction of the drivers, we use driver-assistance systems. Drivers can configure their systems in advance, in analogy to a route guidance system. The systems then act autonomously. Thus, software agents are appropriate. Driver-assistance systems are a platform for *driver-assistance agents* (DAA). We also assume that all intersections host an *intersection agent* (IA). While the DAAs represent the interests of the drivers, the IAs represent the interests of the traffic planners/authorities. To ease presentation, we may use 'vehicle' and 'driver-assistance agent' synonymously in what follows.

4 Intersection-Control Mechanisms

This section presents two new mechanisms for intersection control, *Free Choice* and *Clocked*. To do so, we describe a first-in first-out mechanism (*Time-Slot Request*) which is not valuation-aware that will serve as a reference point for our evaluation. This is appropriate because [1] has shown that such a mechanism outperforms traffic lights. We then describe an auction mechanism for intersection control in abstract terms which subsumes the new mechanisms, before going into details.

The mechanisms are robust in the sense that – due to the physical characteristics of the traffic scenario – unexpected events can occur. For example, the vehicle in front may slow down, or communication may take too long. Thus, vehicles can opt out, e.g., by declining a time slot offered.

4.1 Time-Slot Request (TSR)

We call our implementation of the mechanism of [1] *Time-Slot Request* (*TSR*). We adapt and deploy the FIPA Request Interaction Protocol [5] to implement *TSR*.

Protocol. Vehicles which approach an intersection send a `request` to the intersection agent (IA). It contains the earliest point of time to cross the intersection (*desired time-slot beginning*), the *intersection lane* to use and the id of the vehicle in front (*front vehicle id*). The former two are necessary to allocate a time slot. The order of requests does not necessarily reflect the order of vehicles driving on the same lane. Because vehicles cannot overtake, the IA has to preserve their order on a lane. To accomplish this, the IA uses the id of the preceding vehicle.

The IA processes the requests of vehicles in their order of arrival (first-in first-out). The IA checks if the preceding vehicle has already been assigned a time slot. If not, it cannot assign a time slot to the requesting vehicle. Otherwise the order of time slots may become different from the one of vehicles on the lane. Thus, it refuses the request (`refuse`). In this case, the vehicle has to repeat the request. If the preceding vehicle has already been assigned a time slot, the IA computes the next free time slot for the lane requested which is not earlier than the desired beginning and the time slot of the preceding vehicle. The IA proposes this time slot to the vehicle (`agree`).

If this message takes too much time, or the vehicle in front has slowed down surprisingly, the vehicle might not be able to use the time slot proposed. If so, it sends a `refuse`, otherwise an `agree`. In case of `agree`, the vehicle can now adapt its speed to arrive at the intersection exactly on time. However, as long as the allocation of the slot is not confirmed, it must not enter the intersection. In case of `agree`, the IA tries to allocate the slot. This may fail if another conflicting allocation has taken place meanwhile. If so, the IA returns `failure`, otherwise it confirms (`inform`). The vehicle can use the time slot only after the confirmation. If the request fails, the vehicle repeats it.

4.2 Auction Mechanisms

Now we present the auction mechanisms. An auction consists of two steps, the contact step and the allocation step. In the contact step, the vehicle contacts the IA and requests a time slot. In the allocation step the IA auctions slots to vehicles which have requested one before. During an auction no other auctions take place.

4.2.1 Contact Step

Vehicles request a time slot (`request`) from the IA. Because vehicles cannot overtake, if the vehicle in front has not yet requested a slot, the IA refuses the request (`refuse`). Otherwise, unlike *TSR*, it does not compute the next free slot immediately. It returns `agree` saying that the vehicle will soon receive a call for proposals. Vehicles which have received a refusal issue another request.

The IA uses the requests to update its limited knowledge of the current traffic situation. It maintains a waiting queue for each incoming lane. It stores the desired time-slot beginning, the front vehicle id, the intersection lane and the incoming lane of each vehicle. It adds the vehicle having issued a request to the queue if the queue is empty, or if the front vehicle reported is last in the queue. Vehicles are removed from the queue after having entered the intersection.

4.2.2 Allocation Step

The IA auctions one slot per auction to the vehicles which have requested a slot in the contact step (Figure 1). Note that, if concurrency is supported, several time slots may be auctioned off for the same or overlapping periods of time. The IA does not necessarily offer the same slot to each vehicle. This is because the intersection lane to use, the duration and the beginning of an appropriate slot depend on the vehicle. The IA only announces an *auction interval* $[T_{auct,b}, T_{auct,e}]$ with beginning $T_{auct,b}$ and end $T_{auct,e}$. It determines lower and upper bounds of the auctioned time slot, i.e., the time slot of the winning vehicle must always be within this interval. The winning vehicle receives the earliest possible time slot within the interval given its current position and its maximum speed, next to other conditions.

A vehicle which could receive a slot in the next auction is a *candidate*, i.e., a vehicle which has requested but not yet received a slot, which is the first in its queue without time slot, and for which a possible slot in the auction interval exists. A *candidate time slot* is the earliest possible slot in the auction interval for a candidate.

A time slot must be later than the slot of the preceding vehicle. If the next free time slot begins earlier than the auction interval, the IA computes the earliest free slot within the interval. If there is no such a time slot, the vehicle is not a candidate and is not invited.

Example 2 *Let the auction interval be from* $T_{auct,b}$*=09:45:00 to* $T_{auct,e}$*=09:45:15, and let two candidates* C_1 *and* C_2 *take part in the auction.* C_1 *is a right turning vehicle and needs* 6 *seconds to cross the intersection.* C_2 *is a left turning vehicle and needs* 12 *seconds. Because the intersection lane*

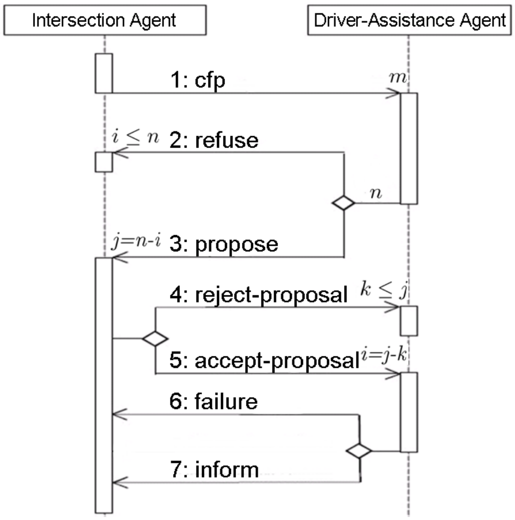

Figure 1: Contract Net Interaction Protocol

of C_1 *is reserved for other vehicles for the first four seconds of the auction interval, the candidate time slot of* C_1 *would be from 09:45:04 to 09:45:10. If* C_2 *won, its candidate time slot would be from 09:45:00 to 09:45:12.*

Several exogenous parameters define the auctions. The IA has to decide when to start the next auction (*auction start time*, T_{start}). Executing an auction takes time, and vehicles need some time to reach the beginning of the intersection and to accelerate. The respective parameter is the *maximum reaction time* ($t_{react,max}$). It determines the minimum difference between auction start time T_{start} and the beginning of the auction interval $T_{auct,b}$. In the real world, this would not be an exogenous parameter, but it is one when looking at our experimental platform in isolation. Section 5 will discuss our choice of parameter values. Further, different auction-interval lengths are conceivable, as we will explain.

Auction Protocol. We use the FIPA Contract Net Interaction Protocol [4] (Figure 1) for the auctions.

The IA sends all m candidates a call for proposals (`cfp`, 1) with the auction interval and their candidate time slot attached. n is the number of candidates replying. If a candidate cannot keep the time slot, i.e., cannot reach the intersection in time, it sends a refusal (`refuse`, 2), otherwise a proposal (`propose`, 3). The proposal contains the *offer*, i.e., the money one is willing to pay for the slot. The IA receives i refusals and j proposals. Thus, if $j > 0$, the IA identifies the best one and the price the candidate has to pay. If there are more than one best proposal, one of them is chosen randomly. The IA then allocates the slot and sends an `accept-proposal` (5) ($l = 1$) to the winner. It sends a `reject-proposal` (4) to all other $k = j - l$ candidates which have sent a proposal. From now on, the IA cannot auction off time slots which conflict with the current one.

If a vehicle receives a reject, it waits for further calls for proposals. When receiving an accept, it checks if it can still keep the slot. If not, it sends a `failure` (6) and waits for another call. Otherwise, it sends back an `inform` (7). The vehicle can now adapt its driving speed to its time slot.

In the failure case, the IA undoes the allocation of the slot so that it can be auctioned off again. The vehicle will receive further call for proposals in the future.

Discussion. We use a sealed-bid auction. Although a first-price auction could be used, we use a second-price auction for the price determination as used in Internet auctions [10]. The auction type determines the bidding strategies, which are beyond the scope of this paper and future work.

In the protocol used, the IA allocates the time slot before sending the `accept-proposal`. If the vehicle replies `failure`, the IA has to undo the allocation. An alternative might be to allocate the time slot only after the vehicle has confirmed sending `inform`. But this alternative is problematic: if the inform message gets lost or takes too much time, the vehicle would use a time slot which has not been allocated yet. This could lead to accidents.

4.2.3 Free Choice

With *Free Choice*, the auction interval is a window moving along the time axis. The IA lets the vehicle that has just won the auction choose its time slot within this interval. It seems obvious to have an interval with fixed duration and to move it along the time axis continuously. However, this has a drawback: The end of the interval changes continuously, and all vehicles may become candidate one after another. There would be only one vehicle participating in each auction, at least when the execution time of auctions is negligible, and if they take place continuously. Namely, vehicles having arrived earlier would already have obtained a time slot. *Free Choice* would be equal to TSR. To avoid this, the end of the auction interval $T_{auct,e}$ moves along in a discrete sequence of steps of fixed length (t_{step}). The effect of this discretization is that it is likely that there are now several vehicles competing for a slot in the new auction interval. Formula 2 below yields the end of the auction interval at the current point of time. Its length is not fixed, since its begin point ($T_{auct,b}$) moves forward continuously. Formula 1 yields this begin point. The auction interval has a minimal length, referred to as $t_{int,min}$.

We distinguish between *successful* and *unsuccessful auctions*. An auction is unsuccessful if the IA receives no proposals, or if the winning vehicle replies with a failure. Otherwise, an auction is successful. The *minimal auction gap* T_{gap} is the minimal gap between the auction start times T_{start} of two subsequent auctions if the first one has not been successful. This gap avoids initiating unnecessary auctions, i.e., auctions without any candidates.

If the last auction A_i has been successful, the next auction A_{i+1} is initiated immediately thereafter. Otherwise, the next auction A_{i+1} is initiated after the minimal auction gap T_{gap} has passed. Because of the maximum reaction time $t_{react,max}$ the beginning of the auction interval is as follows:

$$T_{auct,b} = T_{start} + t_{react,max} \tag{1}$$

This means that $T_{auct,b}$ is different for every auction.

The end of the auction interval $T_{auct,e}$ should be within the interval $[T_{auct,b}; T_{auct,b} + t_{approach,min}]$. It remains unchanged for every auction until t_{step} has passed. It is then increased by t_{step} using the following formula:

$$T_{auct,e} = (\lfloor T_{start}/t_{step} \rfloor + 1) \times t_{step}) + t_{int,min} \tag{2}$$

This means that the duration of the auction interval shrinks from $t_{step} + t_{int,min} - t_{react,max}$ to $t_{int,min} - t_{react,max}$, and then its duration is reset again to $t_{step} + t_{int,min} - t_{react,max}$.

Example 3 *Let the minimum auction interval* $t_{int,min}$ *be* $30.0s$*, let* t_{step} *be* $10s$*, and let* $t_{react,max}$ *be* $5s$*. If the auction start time* T_{start} *is* $0s$*, the auction interval begins at* $5s$ *and ends at* $40s$*. Thus, the length of the interval is* $35s$*. Five seconds later the beginning of the interval is* $10s$*, and the end remains unchanged. The length of the interval is now* $30s$*. If the auction starts at* $10s$*, the interval beginning changes to* 15 *and the end to* $50s$*. The length is* $35s$ *again.*

4.2.4 Clocked

With *Clocked*, the duration of the auction interval is fixed. It is always set to the *maximal time-slot duration*, i.e., the time the vehicle with the smallest acceleration needs to cross the intersection on the longest lane and accelerating from zero to the speed limit. When there are no more candidates for the current auction interval, the interval beginning at the end of the previous auction interval becomes the new auction interval.

With *Clocked*, a time slot can only overlap with another one if both slots were allocated in auctions with the same auction interval. This seems to be a restriction. However, we deem *Clocked* worth to be investigated: These restrictions might be helpful when we combine auctions with mechanisms which allow trading time slots won in an auction before.

Clocked features one further exogenous parameter: the *maximum difference* $t_{offset,max}$ between T_{start} and $T_{auct,b}$.

With *Clocked*, another auction A_{i+1} is initiated immediately after the last auction A_i has completed successfully. In this case the auction interval remains unchanged. Otherwise it changes to the next auction interval.

If a time slot begins too early, the vehicle may not be able to keep it. Thus, we have to avoid auctions whose auction interval is too early. This is the case if:

$$T_{auct,b} < T_{start} + t_{react,max} \tag{3}$$

In this case we set the beginning of the auction interval to $T_{start} + t_{react,max}$.

Auctions with auction intervals too far in the future have to be avoided. This is because no 'interested' vehicle has already entered the neighborhood. We say that the beginning of the auction interval $T_{auct,b}$ is too far in the future if:

$$T_{auct,b} > T_{start} + t_{offset,max} \tag{4}$$

In this case, the IA does not initiate an auction as long as the inequation holds.

This mechanism is called *Clocked* because time slots are clocked with the maximal time-slot duration. This means that several vehicles can cross the intersection at the same time, but only if their time slots are within the auction interval. This does not mean that all time slots must begin at the same time. – Although the duration of the auction interval remains fixed, *Clocked* does not degenerate to *TSR* because the beginning of the auction interval $T_{auct,b}$ as well as the end $T_{auct,e}$ do not change continuously with every auction.

Example 4 *Let the maximum duration of a time slot be* $12s$. *If the auction interval ranges from* $T_{auct,b}$*=00:10:00 to* $T_{auct,e}$*=00:10:12 a right turning candidate may win the slot ranging from 00:10:06 to 00:10:09.*

4.2.5 Subsidies

Both *Free Choice* and *Clocked* have a drawback: Vehicles with high valuations may be stuck behind vehicles with very low valuations. This is because vehicles with very low valuations may loose several auctions in sequence. So far, the vehicles behind it cannot influence this. We overcome this problem by allowing these vehicles to subsidize the bids of vehicles in front.

If subsidies are allowed the IA does not only send the call for proposals to all candidates. Any vehicle which has already requested a slot and is waiting behind a candidate also receives it, and a flag is set indicating that the call for proposals actually is a call for subsidy. These vehicles can then decide to subsidize their front vehicle or not. The subsidy depends on the expected benefit of moving forward one step in the queue. The IA collects bids and subsidies. All bids and subsidies from one lane are added up. The candidate wins which has made a proposal and which has the highest accumulated bid. The price to pay is the second highest accumulated bid. The winning candidate and its subsidizers receive an accept-proposal ($l \geq 1$), all other vehicles receive a refuse-proposal. Each winning vehicle pays a share of the price proportional to its bid.

Subsidies affect the bidding strategies. For instance, candidates might be tempted to offer less than their true valuation hoping for subsidies. Vehicles waiting behind might offer less than their true valuation as a countermeasure.

4.2.6 Further Possible Extensions

The auction mechanisms proposed here do not yet avoid starvation, i.e., in theory, a vehicle might have to wait for an infinite amount of time. The IA can avoid starvation if it interrupts the auctions when the waiting time of a vehicle exceeds an upper bound, lets the vehicle cross the intersection, and resumes. This clearly influences the bidding strategies of the DAAs. [12] indicates that starvation typically is not crucial because the effect on drivers with very low valuations is moderate if subsidies are feasible.

The IA earns money by auctioning time slots to vehicles. This may reduce user acceptance. As long as it keeps this money, the mechanism is not *budget-balanced*. But the IA may return the money earned to the drivers, immediately or after some time, e.g., by reducing taxes. It can return the money according to different schemes, e.g., in proportion to the waiting time. In any case, this extension also affects the bidding strategies.

The auction mechanism might be improved if not only DAAs from the same incoming lane but also DAAs from different incoming lanes can form coalitions, if they do not use conflicting intersection lanes. These coalitions might prevent other DAAs whose intersection lane conflicts with both of their intersection lanes to win the next slot. This extension influences the bidding strategies as well.

5 Evaluation

With our evaluation, we are particularly interested in the average weighted waiting time avgWWT with the various mechanisms, for different demands and degrees of concurrency. We have implemented both *Free Choice* and *Clocked* with subsidies and no other extensions. We always evaluate the mechanisms with subsidies because [12] has shown that subsidies further reduce the avgWWT.

Setting. All experiments share the following settings. The intersection consists of four directions. From each direction there are two incoming and two outgoing lanes. The left incoming lane allows to turn left or to go straight. The right incoming lane allows to turn right or to go straight. The length of each incoming respectively outgoing lane is $800m$. The radius of the intersection is $25m$. For each of the eight incoming lanes vehicles are randomly generated using an exponential distribution with the desired demand as average. Because of the intersection layout each vehicle approaching on a certain incoming lane has two possible routes. Each route is chosen randomly with probability 0.5. After having driven $50m$, the vehicle is allowed to request a slot from the intersection agent. The valuation per second of reduced waiting time is exponentially distributed with mean 0.01. As long as we do not know all realistic bidding strategies and their distribution we assume that each driver always bids and subsidizes his valuation per second of reduced waiting time. In preliminary experiments, we have examined the influence of lower bids and subsidies with *Free Choice*. The bids were randomly distributed between 80% and 100% and subsidies between 50% and 90% of the valuation per second. Because we did not detect a significant effect, we do not consider lower bids and subsidies in the following experiments. All vehicles enter the traffic net with $50km/h$, which is also the speed limit assumed here. We assume that there are different vehicle types, which are equally distributed, and whose possible accelerations range from $1.4m/s^2$ to $3.0m/s^2$. All vehicles decelerate with $3.0m/s^2$.

The simulation is space-continuous and time-discrete, i.e., consisting of subsequent steps. Based on preliminary experiments, the duration of each simulation step is $400ms$. In each simulation run 63 minutes are simulated. To avoid startup effects, we ignore the vehicles which leave the traffic net in the first three minutes.

Because the start time, the vehicle type, the route and the valuation of each driver are chosen randomly, we always execute five simulation runs with the same five seeds for each setting. This lets us compare the i^{th} simulation run of one setting to the i^{th} simulation run of another setting. By doing so, we always compare the same situation, i.e., the same vehicles with the same route and the same valuation.

Calibration of Free Choice. In preliminary experiments we identify appropriate configurations for *Free Choice*. It offers four parameters to optimize: the time after which the end of the auction interval changes (t_{step}), its minimum duration ($t_{int,min}$), the maximum reaction time ($t_{react,max}$), and the minimum auction gap (T_{gap}). We identify adequate values for t_{step} and $t_{int,min}$ in the following experiment. We set the minimum auction gap T_{gap} to the duration of a simulation step, i.e., $400ms$. Preliminary simulations have shown that the value $5s$ for $t_{react,max}$ is sufficient.

If the minimum auction interval $t_{int,min}$ exceeds the minimum approaching time $t_{approach,min}$ every vehicle participates in an auction immediately after requesting a time slot. Since this may lead to many auctions with only one participant, we decided to evaluate different values for the minimum auction interval $t_{int,min}$ and for the period of time after which the end of the auction interval changes (t_{step}), as follows: Because of the speed limit of $50km/h$ and the maximum request distance of $800m - 50m = 750m$ the minimum approaching time $t_{approach,min}$ is $54s$. Thus, we examine the average (weighted) waiting time using combinations of 25%, 50%, 75%, and 100% of $54s$, i.e., $13.5s$, $27.0s$, $40.5s$, and $54.0s$ for both parameters evaluated. We have 16 settings because we simulate four times four combinations. We always use *conflict area exclusive* and set the demand to 250. For each setting we execute five simulation runs and compute the mean of the average (weighted) waiting times. Figures 2 and 3 show these mean values for all settings. The x-axis is the minimum auction interval $t_{int,min}$. The different curves stand for different values of t_{step}.

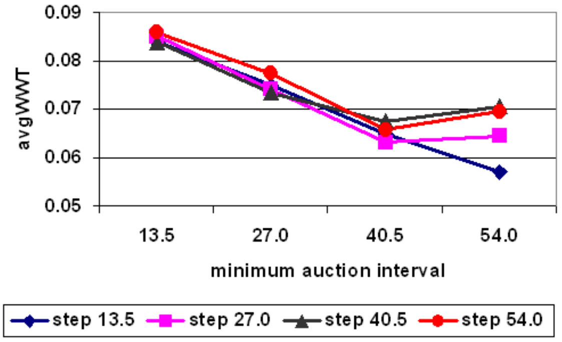

Figure 2: Average weighted waiting time (avgWWT) of Free Choice in different settings

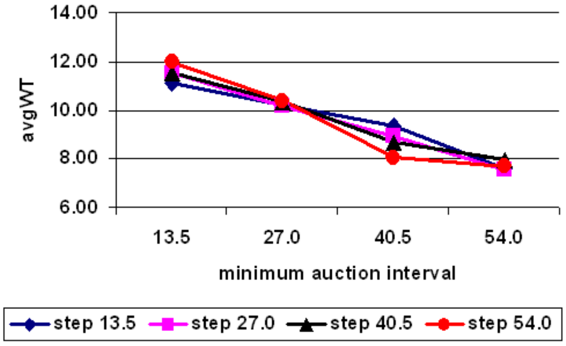

Figure 3: Average waiting time (avgWT) of Free Choice in different settings

The experiment shows that higher values of $t_{int,min}$ decrease both the avgWT and the avgWWT. The influence of t_{step} is rather limited. For minimum auction intervals lower than $54s$ the differences of the average weighted waiting times are negligible for different values of t_{step}. In the case of $t_{int,min} = 54s$ the value

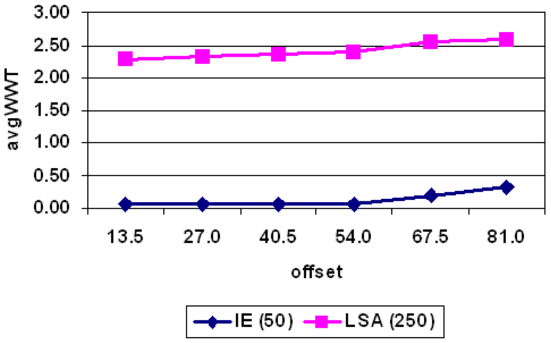

Figure 4: Average weighted waiting time (avgWWT) of Clocked in different settings

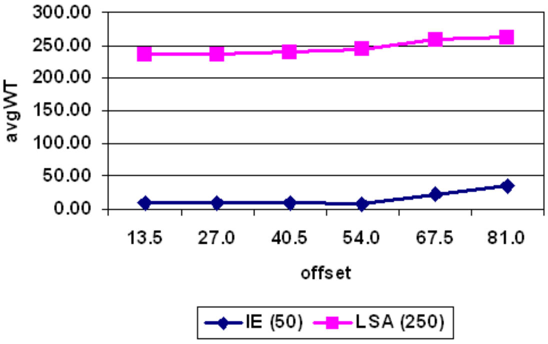

Figure 5: Average waiting time (avgWT) of Clocked in different settings

$13.5s$ for t_{step} leads to the minimal avgWWT of 0.057. We use these parameter settings in the next experiment.

Calibration of Clocked. We now identify good configurations for *Clocked*. *Clocked* has two parameters to adjust: the maximum reaction time $t_{react,max}$ and the maximum difference between auction start time and the beginning of the auction interval $t_{offset,max}$. We set $t_{react,max}$ to $5s$, according to preliminary experiments. In two treatments, we evaluate the effect of the following values for $t_{offset,max}$ on the average (weighted) waiting time: 25%, 50%, 75%, 100%, 125% and 150% of the minimum approaching time $t_{approach,min}$, i.e., $13.5s$, $27.0s$, $40.5s$, $54.0s$, $67.5s$, and $81.0s$. In the first treatment we use *intersection exclusive* and a demand of 50, in the second one we use *conflict area exclusive* and a demand of 250. For each setting we execute five simulation runs and compute the mean of the average (weighted) waiting times. Figures 4 and 5 show these mean values for all settings.

In the treatment with *intersection exclusive* both the avgWT and the avgWWT do not change if $t_{offset,max}$ increases up to 100% of the minimum approaching time $t_{approach,min}$. For the treatment with *conflict area exclusive* we observe a continuous increase. If the values exceeds 100% of $t_{approach,min}$ in both treatments the avgWWT increases significantly. This is not surprising because most vehicles will take part in the auction immediately after requesting a time slot. Thus, the auctions mostly have only one participant, and the valuation is not taken into account. This means that $t_{offset,max} = 13.5s$ leads to the lowest avgWWT. We set $t_{offset,max}$ to this value in what follows.

Treatments. We now compare the average (weighted) waiting times of *Free Choice* and *Clocked* to the ones of *TSR*. We execute one treatment for each degree of concurrency for each mechanism. In the *intersection exclusive* (*IE*) treatment we simulate a demand of 25, 50, and 75. In the *lane exclusive* (*LE*) treatment we simulate a demand from 25 to 225 in steps of 25. In the *conflict area exclusive* (*CAE*) treatment we simulate a demand from 25 to 300 in steps of 25. The difference is due to the fact that the maximum throughput of the intersection depends on the different degrees of concurrency. If the demand gets near or exceeds the maximum throughput of the intersection the waiting queues at the incoming lanes just keep becoming longer. Obviously, this increases both waiting time and weighted waiting time. Thus, we omit higher values in Figures 6, 7, and 8. We expect both *Free Choice* and *Clocked* to perform best for *CAE* independent of the demand. Next to the absolute avgWWT we also compute the relative differences of the avgWWT of *Free Choice* and *Clocked* compared to *TSR*.

***Free Choice* vs. *TSR*.** We plot the avgWWT for *CAE* in Figure 6 and for *LE* in Figure 7. The values for the three variants of *IE* are very similar, so we only plot one curve to avoid clutter in the figure. Further, we use a logarithmic scale for the ordinate. To ease comparisons of the values for the different degrees, both figures contain all curves for *TSR*. The evaluation shows (Figures 6 and 7) that the avgWWT is lower for *Free Choice* (FC) than for *TSR* for all degrees of concurrency and for all demands. This means that *Free Choice* is always effective. The avgWWT for *CAE* is always lower than for *LE*, which in turn is lower than the one for *IE*. This result is independent of the mechanism used with any

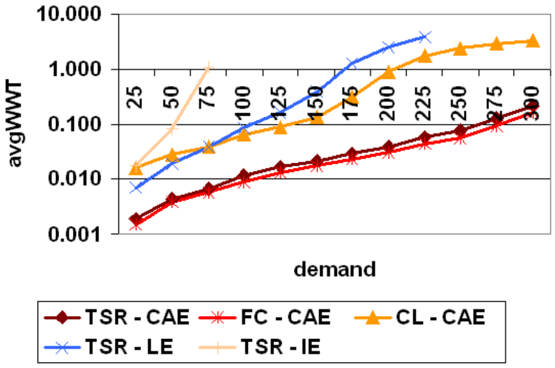

Figure 6: Average weighted waiting time (avgWWT) using conflict area exclusive (CAE)

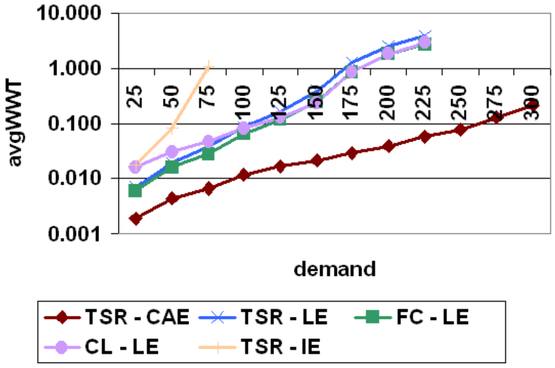

Figure 7: Average weighted waiting time (avgWWT) using lane exclusive (LE)

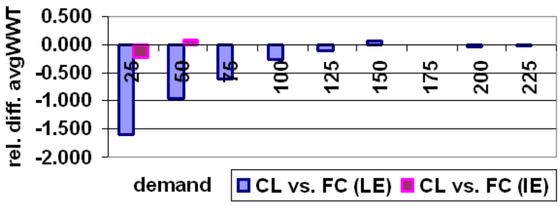

Figure 8: Relative difference of average weighted waiting time (avgWWT) of Clocked vs. Free Choice

degree of concurrency.

The highest relative difference for *CAE* occurs for the demand of 250 (average: 27.8%, 0.95 confidence interval (ci): [22.6%,31.8%]). For *LE* it occurs for the demand of 150 (average: 38.1%, 0.95 ci: [37.8%,38.3%]). With *IE*, the highest relative difference occurs for the demand of 60 (average: 34.5%, 0.95 ci: [22.0%,40.5%]). In the latter case, we have evaluated all demands from 25 to 75 in steps of 5. Note that this does not mean that, say, *LE* performs better for the demand of 150 than *CAE*. The relative difference is computed using *TSR* with the same degree of concurrency. This means that *CAE* should always be used when possible.

The average waiting time avgWT for *Free Choice* is – in the worst case – only slightly higher than for *TSR*. Note that a slight increase of the average waiting time is acceptable because the average weighted waiting time is reduced.

***Clocked* vs. *TSR*.** Figures 6 and 7 show the avgWWT of *Clocked* (CL) and *TSR* for both *CAE* and *LE*. For *CAE* it is always higher with *Clocked* than with *TSR*. Thus, *Clocked* is not effective for *CAE*. For *LE Clocked* is effective for a demand from 100 to 225. For *IE* it is effective only for a demand of 50 and 75.

For *LE* the highest relative difference occurs for the demand of 150 (average: 41.8%, 0.95 ci: [16.9%,53.2%]). Using *IE* the highest relative difference occurs for the demand of 60 (average: 40.0%, 0.95 ci: [31.7%,44.0%]).

***Clocked* vs. *Free Choice*.** We compute the relative difference of avgWWT of *Clocked* compared to *Free Choice* in Figure 8. Because *Clock* is not good at all with *CAE* we omit it from this figure. For *LE* the figure shows that *Free Choice* outperforms *Clocked* for demands below 150. For higher demands the difference is not significant. For *IE Free Choice* is significantly better for a demand of 25 (average: -23.3%, 0.95 ci: [-75.2%,-6.6%]). For a demand of 50 *Clocked* outperforms *Free Choice* (average: 7.0%, 0.95 ci: [3.9%,9.0%]). For 75 the difference is not significant.

Discussion. It has been expected that higher degrees of concurrency reduce the average weighted waiting time. The restriction of *Clocked* that slots from different auction intervals may not overlap affects its performance more for higher degrees of concurrency and lower demands. Except for *CAE Clocked* catches up compared to *Free Choice* with increasing demand. This might be because *Free Choice* yields gaps between time slots more often which are too small for other vehicles.

6 Conclusions

This paper has proposed two novel auction mechanisms for traffic control at intersections, *Free Choice* and *Clocked*. They are the first mechanisms which are both valuation-aware and allow for a realistic, i.e., concurrent, usage of road intersections. *Free Choice* is always effective, i.e., it reduces the average weighted waiting time avgWWT by up to 38.1%. *Clocked* is only effective for lower degrees of concurrency and for higher demands.

In the future, we will examine bidding strategies and the combination with bilateral exchanges of time slots.

References

[1] K. Dresner and P. Stone. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. In *AAMAS 2004*.

[2] K. Dresner and P. Stone. Multiagent Traffic Management: An Improved Intersection Control Mechanism. In *AAMAS 2005*.

[3] K. Dresner and P. Stone. Human-Usable and Emergency Vehicle-Aware Control Policies for Autonomous Intersection Management. In *Fourth International Workshop on Agents in Traffic and Transportation (ATT)*, 2006.

[4] FIPA – Foundation for Intelligent Physical Agents. FIPA Contract Net Interaction Protocol Specification (SC00029H). `http://www.fipa.org/specs/fipa00029/SC00029H.pdf`.

[5] FIPA – Foundation for Intelligent Physical Agents. FIPA Request Interaction Protocol Specification (SC00026H). `http://www.fipa.org/specs/fipa00026/SC00026H.pdf`.

[6] J. T. Greschner and H. E. Gerland. Traffic Signal Priority Tool To Increase Service Quality And Efficiency. In *APTA Bus & Paratransit*, 2000.

[7] G. Jonker, J.-J. Meyer, and F. Dignum. Towards a Market Mechanism for Airport Traffic Control. In *Progress in Artificial Intelligence – 12th Portuguese Conference on Artificial Intelligence*. Springer, 2005.

[8] V. Krishna. *Auction Theory*. Academic Press, 2002.

[9] X. Mao, A. ter Mors, N. Roos, and C. Witteveen. Coordinating Competitive Agents in Dynamic Airport Resource Scheduling. In *Multiagent System Technologies – 5th German Conference*. Springer, 2007.

[10] A. Ockenfels and A. E. Roth. Late and multiple bidding in second price Internet auctions: Theory and evidence concerning different rules for ending an auction. *Games and Economic Behavior*, 55(2):297–320, 2006.

[11] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[12] H. Schepperle and K. Böhm. Agent-Based Traffic Control Using Auctions. In *Cooperative Information Agents XI*. Springer, 2007.

[13] H. Schepperle, K. Böhm, and S. Forster. Traffic Management Based on Negotiations between Vehicles – a Feasibility Demonstration Using Agents. In *Ninth Workshop on Agent Mediated Electronic Commerce (AMEC IX)*, 2007.